AD 749052

NAVAL POSTGRADUATE SCHOOL

Monterey, California

D D C

OCT 4 1972

D

THESIS

IMPROVING PERFORMANCE OF AN IBM 360/67
COMPUTER BY USING A
HARDWARE MONITOR ON THE DISK FACILITY

by

Jay Woodrow Sprague

Thesis Advisor: G. H. Syms

June 1972

Approved for public release; distribution unlimited

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

51

Improving Performance of an IBM 360/67 Computer by using a Hardware Monitor on the Disk Facility

by

Jay Woodrow Sprague
Lieutenant, United States Navy
B.S., United States Naval Academy, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1972

Author Jay Woodrow Sprague

Approved by:

Thesis Advisor

Second Reader

R. E. Gaskell
Chairman, Department of Mathematics

Academic Dean

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Improving Performance of an IBM 360/67 Computer by using a Hardware Monitor on the Disk Facility

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*

Master's Thesis; June 1972

5. AUTHOR(S) *(First name, middle initial, last name)*

Jay Woodrow Sprague

6. REPORT DATE June 1972	7a. TOTAL NO. OF PAGES 51	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

The provision of comprehensive services by a complex modern computer installation is expensive. In the face of increasing demand for computer service, system expansion may be proposed. This expansion may not be necessary if existing resource utilization can be increased or more equally distributed.

This research investigates the possibility of increased system throughput through a balancing of the demand on the individual modules of an IBM 2314 Disk Facility. The performance of the disk modules is measured utilizing a hardware monitor. The hardware monitor is also used to obtain system performance profiles.

Comparison of system throughput is made during times when different sets of resources are available. Recommendations are made to improve system performance by rearranging the data sets on the disk modules.

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Hardware Monitors						
Computer Monitors						
Computer Performance Evaluation						
Performance Monitors						
Optimization Computers						
Optimization Monitors						

Security Classification

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The high cost of providing comprehensive services in a modern computer installation motivates the manager to reduce the costs or at least maintain costs at a constant level during a time of continually increasing demands for service. If the computer system is not completing its assigned tasks according to the required schedule, expansion of the system capacity may be proposed. For example, additional core storage may be added, a faster, greater capacity auxiliary storage device may be substituted for an existing device, faster input/output peripheral devices may be obtained, or even the CPU itself may be upgraded. All of these alternatives involve significant financial expenditure, but another alternative exists which may be less expensive. That is to continue to utilize the same equipment, but to increase the effective utilization of this equipment to meet the increased demand for computational power.

In order for this last alternative to be selected, its feasibility must be determined and before making this determination one must measure the present performance of the system.

The measurement of the performance of a complex computer system is difficult, but the potential rewards are significant and well documented (References 1 and 2). In addition to the improvements in performance made possible by performance measurement, the measurements provide a basis for future decisions on configuration changes and system expansion.

This research is directed at the performance measurement of an IBM 2314 Disk Facility and its associated selector channel. The performance of individual disk modules is measured and the resulting data is analyzed. Recommendations are presented to improve system performance.

II. BACKGROUND AND OBJECTIVES

Knowing what to measure with a hardware monitor is difficult. If measurements are too gross, specific recommendations for change are difficult or impossible to formulate, while if measurements are very detailed they may form the basis for a recommendation to improve utilization of one component of a system without considering the concurrent effects on the overall system performance.

This research is part of a continuing effort to determine precisely how to identify and measure the work accomplished by and performance of a complex modern computing system. The very definitions of the terms "work", "performance", and "computer power" are under discussion and subject to efforts for more formal definition (Reference 3).

Hanke has made measurements on the IBM 360 Model 67 installed at the Naval Postgraduate School (Reference 4). He reported a large percentage of CPU wait only time (CPU in wait state and selector channel 2 not busy). One possible cause of this is large disk arm seek time, i.e., the CPU and selector channel are both waiting for the disk arm to move to some other disk track. The primary objective of this research was to measure the performance of the IBM 2314 Disk Facility and its associated selector channel to determine the percentage of time spent by the CPU and selector channel both waiting for the disk arm to move to another track, then to determine if this arm seek time accounted for the majority of the CPU wait only time. In addition to specific measurements of the 2314 Disk Facility, it was desirable to record broad system performance profiles to determine if CPU wait only continued to be significantly high.

Some improvements were recommended in the data reduction and analysis programs written by Hanke (Reference 4). These improvements included the addition of the ability to plot the output from the hardware monitor and the addition of a date check in the program SMF Graph. Some improvements in the statistical analysis of data to determine means, variances and correlations was also required.

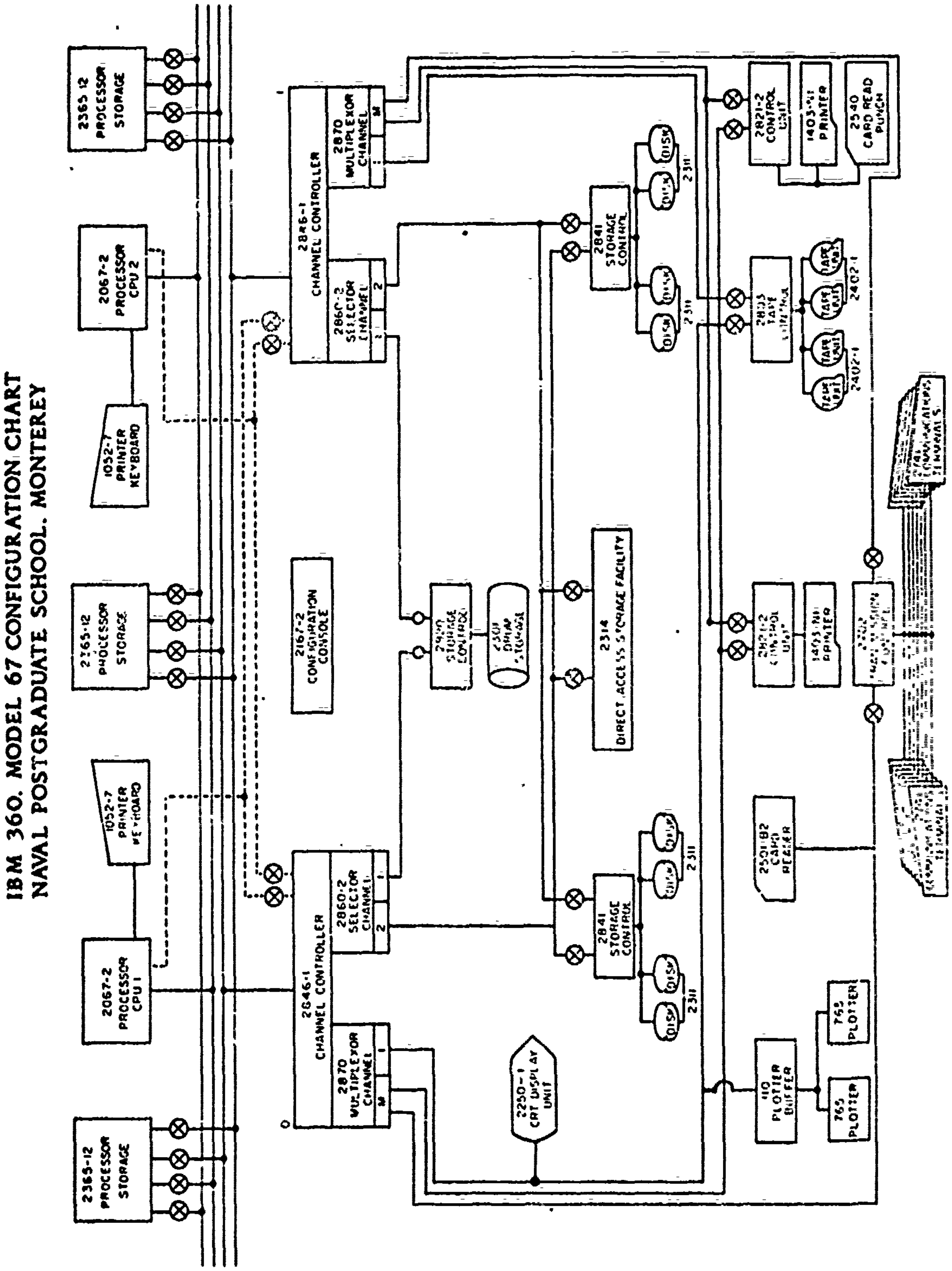
Thus, the overall objective of the research was to combine a specific performance measurement experiment with supporting data reduction and analysis in order to make a specific recommendation for improvement in system performance. The objective of this thesis is to report the results of this research and to suggest areas for further research.

III. EXPERIMENTAL PROCEDURE

A. MEASUREMENT ENVIRONMENT

The computer system under investigation is an IBM 360 Model 67 with configuration as shown in Figure 1. The system was operated in a simplex mode (single CPU) with 768K bytes of core storage for 20 hours per weekday and as a split system (separate operating systems on the two CPU's) for four hours per day. On Saturdays and Sundays the system is run from 0800-2000 in a simplex mode. While a split system is operating from 1200 to 1600 each weekday, part of the computer resources are assigned to a time-sharing system, CP/CMS (Cambridge Monitor System). The major change in resources available for batch processing operation includes the loss of 256K bytes of core storage and the 2301 drum. Detailed allocation of resources during the four hours of time-sharing is shown in Figure 2.

The operating system under investigation is OS/360 MVT (Multiprogramming with a Variable number of Tasks). (The operation of the CP/CMS time-sharing system was not measured as part of this research.) During the twenty hours per day without time-sharing, 768K bytes of core storage are available with 478K bytes available for the execution of problem programs and the remaining 290K bytes for use by the operating system. The use of 256K bytes by the time-sharing system leaves 222K bytes for problem programs during the 1200-1600



Naval Postgraduate School IBM 360 Model 67

Figure 1.

DEVICE	OS/MVT	CP/CMS
CPU 2067-2	X	
CPU 2067-2		X
PRINTER KEYBOARD 1052-7	X	
PRINTER KEYBOARD 1052-7		X
CORE STORAGE 2365-12	X	
CORE STORAGE 2365-12	X	
CORE STORAGE 2365-12		X
DRUM STORAGE 2301		X
DISK STORAGE 2311 (8)		X
DISK STORAGE 2314	X	
TAPE UNITS 2402-1	X	X
CARD READER 2501-B2	X	
CARD READ PUNCH 2540		X
PLOTTER 765 (2)	X	
PRINTER 1403-N1 (2)	X	X
CHANNEL CONTROLLER 2846-1 (2)	X	X

Computer Resource Allocation Under CP/CMS

Figure 2.

time period. (Since this research was completed some parts of the operating system, namely the resident SVC's, have been made non-resident and this increased the usable core to 260K bytes.)

Operating policy also varies with the time of day. The primary objective of the operating policy is to give quick turnaround for small, short jobs (≤ 100K bytes, ≤ 20 seconds CPU time). No particular attempt is made to balance the workload of the system, i.e., control the job mix to execute both I/O bound and compute bound jobs at the same time. Control of the job mix would be difficult as job entry is by way of a user operated hot card reader. Job classes are defined to give the highest priority to the small, short jobs. The use of QUICKRUN (Reference 5) as a sub-system of the operating system is also highly favorable to the small jobs, generally providing "instant" turnaround (less than five minutes) for the small jobs. QUICKRUN is a job management system which processes problem programs faster than OS/MVT by reducing the operating system overhead associated with each job. Restrictions on jobs eligible to be run under QUICKRUN include less than 100K bytes, less than 20 seconds of CPU time, no use of tape, and less than 1000 lines of printed output.

Job arrivals are heavily concentrated in the afternoon with the peak load usually coming between 1400-1600. During the month of March 1972 when these measurements were taken, 24,500 jobs were processed; of these 11,700 were under QUICKRUN.

B. EXPERIMENTAL DESCRIPTION

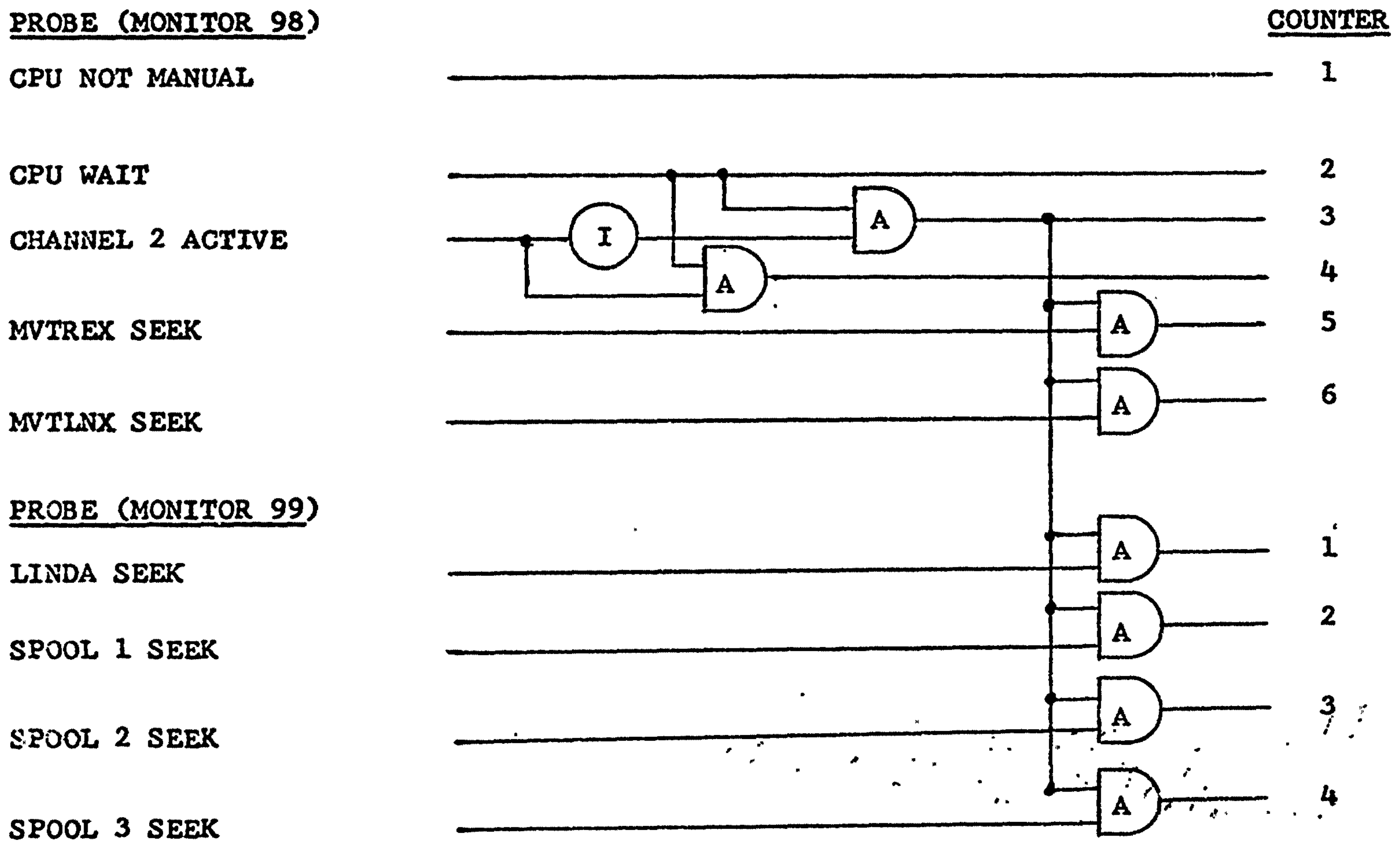
The primary measuring device used in this research was the Measurement Engine, a hardware monitor manufactured by Boole and Babbage, Inc.. The use of the Measurement Engine in system performance measurement and analysis is described in References4 and 6. The Measurement Engine is actually a hardware monitor system with many different possible configurations. As used for these experiments, the configuration consisted of two ME-1011 Event Monitors and one ME-2011 Paper Tape Printer, all owned by this institution. Each Event Monitor can receive signals from eight probes attached to the host computer. The probe signals may then be combined on a user wired logic plugboard which has AND, NOR, INVERTER, and FLIPFLOP capabilities. The outputs from the logic plugboard are then routed to the six counters and the paper tape printer. Logic signals may be routed between Event Monitors which may be stacked one upon the other.

To obtain the nine signals shown in Figure 3, nine probes were connected to the appropriate computer pins also shown in Figure 3.

SIGNAL	DEVICE	PIN
CPU manual	2067	EC2H4B09
CPU wait	2067	EC2J6B07
Channel 2 busy	2360	BA3D6D04
MVTREX disk arm seek	2314	AA3H4D11*
MVTLNX disk arm seek	2314	AA3H4D11
LINDA disk arm seek	2314	AA3H4D11
SPOOL 1 disk arm seek	2314	AA3H4D11
SPOOL 2 disk arm seek	2314	AA3H4D11
SPOOL 3 disk arm seek	2314	AA3H4D11

*This pin is probed on each module measured.

Figure 3. Hardware Monitor Signal Probe Connections



Logic Diagram

Figure 4.

The input signals were combined using the logic board capability of the Event Monitor. A diagramatic representation of the logic is shown in Figure 4. The resulting signals representing the ten events shown in Figure 5 were accumulated by the counters of the two Event Monitors and at preselected time intervals were recorded by the Paper Tape Printer. These paper tape data were then keypunched to be used as input to the program Hardware Graph (Reference 4), which presents a bar graph for each event for each time interval.

1. CPU not manual
2. CPU wait
3. CPU wait and selector channel 2 busy
4. CPU wait and selector channel 2 not busy
5. CPU wait and selector channel 2 not busy and disk module MVTREX arm seeking
6. CPU wait and selector channel 2 not busy and disk module MVTLNX arm seeking
7. CPU wait and selector channel 2 not busy and disk module LINDA arm seeking
8. CPU wait and selector channel 2 not busy and disk module SPOOL 1 arm seeking
9. CPU wait and selector channel 2 not busy and disk module SPOOL 2 arm seeking
10. CPU wait and selector channel 2 not busy and disk module SPOOL 3 arm seeking

Events Monitored.
Figure 5.

The conditions of each experiment are summarized in Figure 6, but it is appropriate here to discuss some of the reasons for conducting the experiments under these conditions.

Experiment	Week of quarter	Day of week	Time of day	Size of interval	Number of intervals
1	10	W	1115-1715	15 min.	24
2	10	W	1715-2315	15 min.	24
3	10	Th	0900-2100	30 min.	24
4	10	F	0900-2100	30 min.	24
5	10	Sa	0915-1915	30 min.	20
6	11	Su	0930-1930	30 min.	20
7	11	Tu	1015-2015	60 min.	10

Experiment Summary

Figure 6.

In order to insure robustness of results, it was desired to conduct worst case experiments and analysis. The tenth and eleventh weeks of a twelve week academic quarter were chosen as appropriate times for measurements due to the historically heavy workload during these two weeks.

It was also desired to compare system performance during the time periods when the time-sharing system CP/CMS was being utilized against the periods when OS/MVT was operating exclusively. This dictated that the afternoon be included. Also, the highest job arrival frequency is during the afternoon.

For the first two days' experiments (1 and 2), a time interval of 15 minutes was chosen in order to determine the range of values over relatively short time intervals. There were no wide fluctuations during the 15 minute intervals so 30 minute intervals were chosen for the remaining experiments. The 60 minute interval was chosen for the final experiment due to the failure of the paper tape printer. The hardware monitor holds the accumulated utilization values in a buffer for output to the paper tape printer until the next time interval has elapsed. This allows the experimenter to hand record the values in the buffer just before the end of a time interval and just after the end of a time interval. By recording data from two time intervals, the experimenter may then by physically absent from the hardware monitor for slightly less than two more time intervals. For example, by using the 60 minute interval, one may be absent from the hardware monitor for about 1 hour and 50 minutes of every 2 hours without losing any data. It is felt that these different time intervals do not significantly affect the results reported herein.

System performance was monitored for a total of 66 hours. Of this time there were 768K bytes of core storage available to the system for 50 hours. For 16 hours 512K bytes of core storage were available to the system as 256K bytes of core storage and the 2301 drum were being utilized by the time-sharing system. The 66 hours of measurement time were divided into 46 hours during weekdays and 20 hours during the weekend.

IV. DISCUSSION OF RESULTS

A. DISK MODULE PERFORMANCE

The six modules of the IBM 2314 Disk Facility whose performance was measured are known by the names MVTREX, MVTLNX, LINDA, SPOOL 1, SPOOL 2, and SPOOL 3. Two other user disk modules named MARY and DUFFY were not measured because their activity is much lower than those measured. In this discussion the comparisons involve the condition when the CPU is in the wait state and the selector channel is not busy and a disk arm is seeking (moving to another track). This condition will be referred to, for example, as MVTREX seek without repeating the CPU wait and channel not busy qualifiers.

System performance profiles (Figure 7) show that the CPU wait percentage had a wide range of variation varying from 0 to 86 percent. Averaged over the seven experiments the mean CPU wait was 51 percent. The CPU wait only (CPU wait and selector channel 2 not busy) averaged over the seven experiments ranged from 6 to 55 percent with a mean of 26 percent. Thus, on the average, the CPU is idle half the time and of this CPU idle time about half the time the channel is also idle. One condition that may cause both the CPU and channel to be waiting is a disk arm seeking (moving to another track). Data from the seven experiments showed that the module MVTLNX had more arm seek time than the other five disk modules measured (Figure 8). The ratio

Machine not manual	max	100	100	100	100	100	100	100	100
	mean	99	100	100	98	100	100	99	100
	min	76	99	97	66	100	100	98	91
CPU wait	max	86	72	82	84	70	77	69	77
	mean	64	40	59	64	35	45	48	51
	min	30	1	27	34	0	8	21	17
CPU wait and channel 2 busy	max	49	30	44	43	27	34	34	37
	mean	32	19	31	31	13	16	27	24
	min	10	0	16	18	0	5	13	9
CPU wait and channel 2 NOT busy	max	61	57	58	61	47	63	37	55
	mean	31	21	27	34	22	29	22	26
	min	13	0	10	12	0	3	7	6
CPU Active	max	70	99	73	66	100	92	79	83
	mean	36	60	41	36	65	55	52	49
	min	14	28	18	16	30	23	31	23
EXPERIMENT		1	2	3	4	5	6	7	Avg.

System Performance Profiles (Percentages)

Figure 7.

CPU wait and	max	6.78	33.73	6.72	4.72	3.52	3.43	4.93	9.12
channel 2 NOT busy	mean	3.11	2.93	3.02	2.70	1.35	1.75	2.59	2.49
and MVTREX seek	min	1.66	0.0	0.89	1.19	0.0	0.61	0.66	0.72
CPU wait and	max	68.02	10.86	14.08	19.47	10.26	9.38	11.47	20.51
channel 2 NOT busy	mean	14.50	6.41	10.36	10.86	4.96	4.63	7.93	8.52
and MVTLNX seek	min	5.06	0.06	5.43	3.51	0.0	0.82	2.89	2.55
CPU wait and	max	0.48	10.05	0.12	1.75	0.06	0.08	1.32	1.75
channel 2 NOT busy	mean	0.06	0.43	0.04	4.21	0.02	0.03	0.18	0.75
and LINDA seek	min	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CPU wait and	max	11.30	1.15	3.04	4.70	2.02	1.06	3.76	3.76
channel 2 NOT busy	mean	2.40	0.52	1.35	1.38	0.51	0.27	5.57	1.71
and SPOOL 1 seek	min	0.39	0.0	0.23	0.19	0.0	0.03	0.0	0.12
CPU wait and	max	11.82	7.97	13.39	9.70	5.79	6.00	8.92	9.05
channel 2 NOT busy	mean	4.87	1.62	4.33	3.35	1.29	1.41	5.97	3.26
and SPOOL 2 seek	min	0.71	0.0	1.38	0.27	0.0	0.30	0.0	0.38
CPU wait and	max	7.19	8.35	10.23	22.59	18.60	6.55	10.72	10.72
channel 2 NOT busy	mean	2.30	2.53	2.82	3.64	2.00	1.20	3.68	2.60
and SPOOL 3 seek	min	0.78	0.11	0.74	0.30	0.0	0.07	0.0	0.29
EXPERIMENT		1	2	3	4	5	6	7	Average

Disk Activity Summary (Percentages)

Figure 8.

of MVTLNX seek to the other disk modules ranged from 1.9:1 to 12.0:1. The ratio of MVTLNX seek to the mean disk seek was 2.65:1 averaged over the seven experiments. Thus, there was an unbalanced demand placed on this one disk module, MVTLNX.

What then were the contents of this disk module which may have caused this imbalanced demand? MVTLNX is a system module with three particular data sets of interest. The most active data set on MVTLNX was the operating system job queue. This job queue data set is allocated 30 cylinders (4.3 million bytes) of space which is referenced by many parts of the operating system. The job queue must be accessed by the reader program, the initiator program, the writer program, and by the display commands issued by the operator's console, for a minimum of between 6 and 16 accesses per job.

Another significant data set on the MVTLNX module is the link library. This data set is allocated 50 cylinders (7.2 million bytes) of space. The link library contains the executable modules for the reader program, the writer program, and the initiator program. It also contains the language processing modules (FORTRAN G, FORTRAN H, PL/I, COBOL, and RPG), non-resident operating system modules, supervisor calls (SVC) and input/output error recovery modules. This data set must be accessed a minimum of 3 to 5 times for each job execution.

The third data set of interest in the module MVTLNX is used for recording accounting data. System Management

Facilities (SMF) information is written into this third data set. SMF is an optional feature of OS/MVT which records system and job performance information. (Use of SMF as a software monitor is explained by Hanke (Reference 4).) In particular, job start and stop times, CPU times used and identification data are recorded for each job step upon completion of the job step. This data set is thus accessed at least three times on an average, non-QUICKRUN job (once per job step).

This one disk module MVTLNX therefore contains three data sets which must be accessed between 12-24 times for each job execution. This would be the case for a typical FORTRAN compile, link-edit and execute job, which account for about half of all jobs submitted, not including those FORTRAN jobs run under QUICKRUN.

The questions arise as to which of the data sets could be transferred from MVTLNX to another location, where the data set could be relocated, and what effect the relocation would have on system performance. The first data set examined was the job queue. The job queue is presently allocated 30 cylinders (4.3 million bytes) of space with a resulting capacity of about 150 jobs. Assuming that the accesses to the job queue account for between 50-67 percent of the disk seek activity on MVTLNX, and that the mean MVTLNX seek is 8.52 percent, then the job queue seek is from 0.50 * 8.52 = 4.3 to 0.67 * 8.52 = 5.7 percent. Considering that 20 hours per day the system is run with no time-sharing (i.e., with the 2301 drum available), then from 52 minutes to 69 minutes (20 hours * 0.057 = 1.14 hours = 69 minutes, 20 hours * .043 = .86 hours = 51.6 minutes) per day is spent waiting for access to the job queue.

Now suppose the job queue were placed on the 2301 drum. Arm seek delay would be nonexistant and although there would be some delay in the form of rotational delay, the delay would be less than the rotational delay of the 2314 disk. The improvement gained would be at most .86/20 hours = 4.3 percent or 1.14 hours/20 hours = 5.7 percent. The 2301 drum is used for four hours per day in support of the CP/CMS time-sharing system and therefore a utility program would be needed to transfer the job queue from the 2314 disk to the 2301 drum and back again at the conclusion of the time-sharing period. This transfer of the job queue would also require a reformatting of the job queue to coincide with the recording techniques used on the 2301 drum. The required utility program does not exist and one NPS system programmer suggested that it would be very difficult to write. Disadvantages in moving the job queue from the disk to drum and back include operator inconvenience, time required for transfer, and possible error and subsequent loss of the job queue.

Another alternative would be to move the job queue to another disk module. Currently there would have to be an examination of the other disk modules to determine which data sets should be moved to make room for the job queue, as there is not sufficient empty space on the other modules to relocate the job queue. The effect on performance would be difficult to estimate, however, this task would have significant value of serving as a basis for comparison with a repeated conduct of the same experiments after the job queue had been relocated. (This possibility is discussed later.)

What other data set might be moved? The link library is currently allocated 50 cylinders (7.2 million bytes) of space which is about twice the capacity of the 2301 drum. Similar comments to those about moving the job queue to another disk module apply to moving the link library to another disk module.

This leads one to consider the System Management Facilities (SMF) data sets. Two data sets, SYS1.MANX and SYS1.MANY are utilized for recording SMF data. Two data sets are used so that when one data set is full the recording is switched to the other data set. Another data set on the disk module MVTLNX is named SYS1.SMFTUB. The data from SYS1.MANX or SYS1.MANY is tranferred to SYS1.SMFTUB as each is filled. Later SYS1.SMFTUB is transferred to magnetic tape. When the transfers from SYS1.MANX or SYS1.MANY to SYS1.SMFTUB take place, the disk arm must move back and forth on the same disk module, the same disk module which already is the most active. This occurs about once per day and the transfer is usually done on the 0000-0800 shift to minimize the effect of disk arm interference on system performance.

Assuming that the SMF recording is 12.5 to 25 percent of the activity on the disk module MVTLNX and using the mean of

8.52 percent MVTLNX seek averaged over the seven experiments, SMF recording would account for .25 * 8.52 = 2.13 percent of MVTLNX seek. Taking 2.15 percent * 20 hours = 0.430 hours = 25.8 minutes per day spent waiting for the disk arm to move to another track in order to record SMF data. Considering that this time also causes cont. .tion with the job queue and link library activity, it would be advantageous to record SMF data on one of the more lightly used disk modules. Elimination of the 25.8 minutes SMF time would represent at most .43 hours/20 hours = 2.15 percent improvement in the activity, performance, or 1.07 percent if SMF recording is 12.5 percent.

If the two suggested changes in the contents of the disk module MVTLNX were made, moving the job queue to the 2301 drum and moving the SMF data sets to a less active disk module, the total improvement would be at best 5.7 percent + 2.15 percent = 7.85 percent improvement. Using an average of 44 job steps executed per hour this 7.85 percent improvement would represent 3.45 additional job steps per hour throughput or 69 additional job steps per 20 hour day.

Two implied assumptions affecting disk seek time that should be explained here are the order of requests to the disk and the distance between active data sets. Since the exact order of requests is unknown and the requests do not follow any fixed pattern in a multiprogramming environment, the assumption of random ordering seems reasonable. The three critical data sets - SYS1.JOBQUE, SYS1.LINKLIB, and

SYS1.MANX(Y) - are located contiguously so as to minimize the arm seeking delay and thus neglecting the actual distance moved and averaging the arm seek times seems reasonable.

B. SYSTEM THROUGHPUT

Discussion of disk performance in particular and computer system performance in general must be considered in the context of system throughput. During the month of March 1972 when these experiments were performed, the computer center processed 24,497 jobs. Of this total number of jobs, 11,681 were run under the QUICKRUN job management system. Figure 9 shows the system throughput in terms of jobs completed during each hour of the day. Considering the period of time from 1200-1600 the average number of jobs completed per hour is 1933 whereas for the next busiest four hours (1000-1200 and 1600-1800) the average number of jobs completed is 1699. If these averages are normalized to reflect the different quantities of problem program core available (222K bytes from 1200-1600 and 478K bytes from 1000-1200 and 1600-1800), then the throughput per unit core is even greater during the 1200-1600 time period while 256K bytes core storage are lost to the time-sharing system. Lest one conclude that a reduced amount of core storage improves system throughput one must consider the different operating policies in effect during these two different time periods.

During the 1200-1600 time period when only 222K bytes of core storage are available, only small, short jobs, 100K bytes or less,20 seconds CPU time or less, are allowed to be

Hour Ending at Time	Jobs	% Total	Hour Ending at Time	Jobs	% Total
0100	434	1.8	1300	1835	7.5
0200	277	1.1	1400	1782	7.3
0300	173	0.7	1500	2051	8.4
0400	150	0.6	1600	2065	8.4
0500	117	0.5	1700	1824	7.4
0600	110	0.4	1800	1727	7.0
0700	104	0.4	1900	1139	4.6
0800	72	0.3	2000	1022	4.2
0900	559	2.3	2100	1139	4.6
1000	1313	5.4	2200	1261	5.1
1100	1749	7.1	2300	1153	4.7
1200	1495	6.1	2400	946	3.9

TOTAL	24,497	100%
QUICKRUN	11,681	47.5%

March 1972 System Throughput

Figure 9.

run. This control is obtained by a combination of two factors. First, job classes are defined to segregate these jobs into one class and secondly, the operator controls the starting of initiator programs to run only this one class of jobs. Thus, the operating policy favors the predominant job type, giving fast turnaround to these jobs and operating within the core storage limitations imposed by the loss of 256K bytes of core storage for use by the time-sharing system. This operating policy discriminates against larger, longer jobs and also has an effect on system utilization. There is not a mix of I/O bound jobs and compute bound jobs during this time period so that CPU utilization decreases while I/O activity increases (Figure 10).

C. PLOTTING AND STATISTICAL ANALYSIS PROGRAMS

Some improvements were recommended in the data reduction and analysis programs written by Hanke (Reference 4). A program, Hardware Graph, processes data from the hardware monitor by reading keypunched data cards and producing bar graphs for each event monitored. It was desired to plot multiple events on one graph so that the analyst might be able to determine trends or possible interaction between various events. The plotting program listed in Appendix A is adapted from the locally obtained program STPLOT. By changing a FORTRAN READ statement and corresponding FORMAT statement, the user may plot various combinations of events, up to a maximum of ten. The plot is output on the line

EVENT	512K bytes no drum	768K bytes and drum	Ratics
CPU wait	65.18	47.13	1.38
CPU wait and channel not busy	29.73	26.07	1.14
CPU wait and channel busy	34.61	21.06	1.65
CPU wait, and channel not busy and MVTREX seek	3.51	2.22	1.58
CPU wait and channel not busy and MVTLNX seek	12.34	7.62	1.62
CPU wait and channel not busy and LINDA seek	2.87	0.12	23.9
CPU wait and channel not busy and SPOOL 1 seek	2.26	1.14	1.98
CPU wait and channel not busy and SPOOL 2 seek	6.03	2.15	2.8
CPU wait and channel not busy and SPOOL 3 seek	2.64	2.52	1.05

Comparison of OS/MVT performance
with 768K bytes vs. 512K bytes of Core

Figure 10.

printer and the user must draw lines to connect the points corresponding to the events plotted. The rapid turnaround for this program makes it very useful for quick visual analysis of experimental results.

Hanke's program, SMF Graph, reads the System Management Facilities data from the SMF data set on the disk module MVTLNX and provides a summary and some analysis of this job stream data. One input parameter to this program is time of day when measurement starts. This is adequate to locate the desired SMF data if data from only one day is currently recorded. Sometimes data from more than one day is in the SMF data set in which case the desired data might not be obtained using the original version of SMF Graph. An assembly language subprogram was added to SMF Graph to require the user to input the desired date as another input parameter to SMF Graph and to give SMF Graph the capability to check for that date in the SMF data set.

Statistical analysis of the hardware monitor output was performed with the assistance of programs from UCLA's BIMED series (Reference 7). These programs provide many standard statistical measures such as means, variances, correlations with a minimum effort on the part of the user. An example of the results of computation for one experiment is shown in Appendix C.

D. FIGURE OF MERIT

During the course of this research, the question arose as to whether the results obtained were typical of those which might be obtained by similar experiments on other computer systems. Also what experiments in measuring computer performance are in progress at other university computing centers? Estrin in Reference 8 states that the results of experiments should be reproducible in order to be of any value for subsequent generalization.

For these and other reasons, a survey was designed to inquire about the computer performance at other computer facilities. Shown in Appendix D, this survey will be sent to many installations which use an IBM 360/67 and to many other universities. The results will be compiled and made available to contributors in an effort toward further understanding of computer performance measurement and computer system performance optimization.

One key question in the survey asks, "Is there any one overall figure of merit or performance index computed by combination of several performance parameters? (Please give formula)". The possibility of obtaining a concise answer to this question seems sufficiently remote since very little research has been done on this problem, although this question is currently under study at this institution. If there is a valid figure of merit for a computer installation, or a computer operating environment, it would certainly be of interest and of value to other computer center staffs.

V. CONCLUSIONS AND RECOMMENDATIONS

There are three positive results derived from the conduct of this research. First, the actual performance of the computer system during a stipulated time period can be stated as a fact rather than a conjecture; this can be used as a basis for future performance comparisons. Secondly, a positive recommendation for improvement can be made and thirdly, the author is now prepared to conduct further performance evaluation analyses of computer systems.

The ability to state the performance of a computer system as a fact is valuable to the manager of a computer system. Plans and decisions can be based on this factual performance data with some level of confidence, which is certainly greater than the confidence based on unproven conjectures. In addition, future performance measurements can use the results reported here as a basis for comparison. Any comparison, however, would have to carefully reconsider the measurement environment.

The ability to make a positive recommendation is particularly significant. It may be very interesting to measure performance of various components of a computer system, however if no positive recommendation for improvement can be made the effort expended in measurement is wasted. The recommendation from this research is to move the operating system job queue data set and the System Management

Facility (SMF) recording data set to a more lightly used disk module on the 2314 Disk Facility. Using three disk modules for storage of operating system data sets would balance the demand on the individual disk modules. Moving the job queue to a third system disk module could lower the mean seek wait on MVTLNX by 2.82 percent (8.52 - 5.7) (Section IV, A). This would result in a 2.9 percent increase in system throughput (2.82/100-2.82) plus some additional increase due to the elimination of arm seek contention on the disk module MVTLNX. Balancing the demand on the individual disk modules is therefore estimated to represent a 3 to 5 percent improvement in system throughput.

The computer system at the Naval Postgraduate School is owned by the U. S. Navy. Using the replacement cost of $4.8 million and an estimated 60 months (5 years) of system life, a monthly lease cost of (4.8 million/60 months) $80,000 may be assumed. A 3 to 5 percent improvement thus represents a $2400 to $4000 potential savings. A $2400 to $4000 monthly savings would pay for the cost of the hardware monitor used for the performance measurements in less than 4 to 8 months time. Thus, this one experiment in performance improvement, by paying for the hardware monitor, provides the potential for future performance measurement efforts at essentially no cost.

Concurrent with the reporting of this research, a later version of the operating system known as Release 20 of OS/MVT is being implemented at this computer center. A decision has been made to eliminate the use of the disk module name LINDA as a user disk module and to use LINDA as a third operating system disk module. Thus, the results of performance measurement are providing an input to the decision making process for configuration changes. It is important to suggest that measurements be taken to verify the suggested improvement in system performance and to determine if the new version of the operating system has created any previously unknown problems.

The preparation and education of the author to conduct future performance evaluation analyses of computer systems is a result of this research effort which may be of real benefit to the Navy. The number of trained analysts in computer system performance evaluation is small in contrast to a growing need. It does not appear that main frame manufacturers are going to expend great effort to assist clients in performance optimization through performance measurement as this would probably reduce sales of additional equipment. The users therefore will have to train their own performance analysts or resort to outside consultants in order to use performance measurement to optimize system resource utilization.

Further performance measurement of the computer system at this installation would be useful. Questions requiring further research include:

1. What part of the CPU wait only time is spent waiting for an operator's console response.

2. Does the operator's console activity vary widely from shift to shift?

3. How has the addition of the IBM 2321 Data Cell affected system performance?

4. What is the effect of having non-resident Supervisor Calls (SVC's) when only 512K bytes of core storage is available?

5. What other parts of the operating system could be made non-resident?

In addition to performing specific performance measurement experiments, it is recommended that this computer installation establish a plan for periodic system profile measurements. Monthly accounting data is currently recorded and presented to the analyst in a very usable form. The same amount of effort should be expended to provide monthly hardware performance profile information to accompany the accounting data.

This thesis describes the steps taken to improve the performance of a computer system. Further improvements in performance may be available for the cost of performing further analysis. Since each one and a half percent improvement amounts to $1200 per month increase in computing power for the rest of the system life, these improvements should be actively pursued.

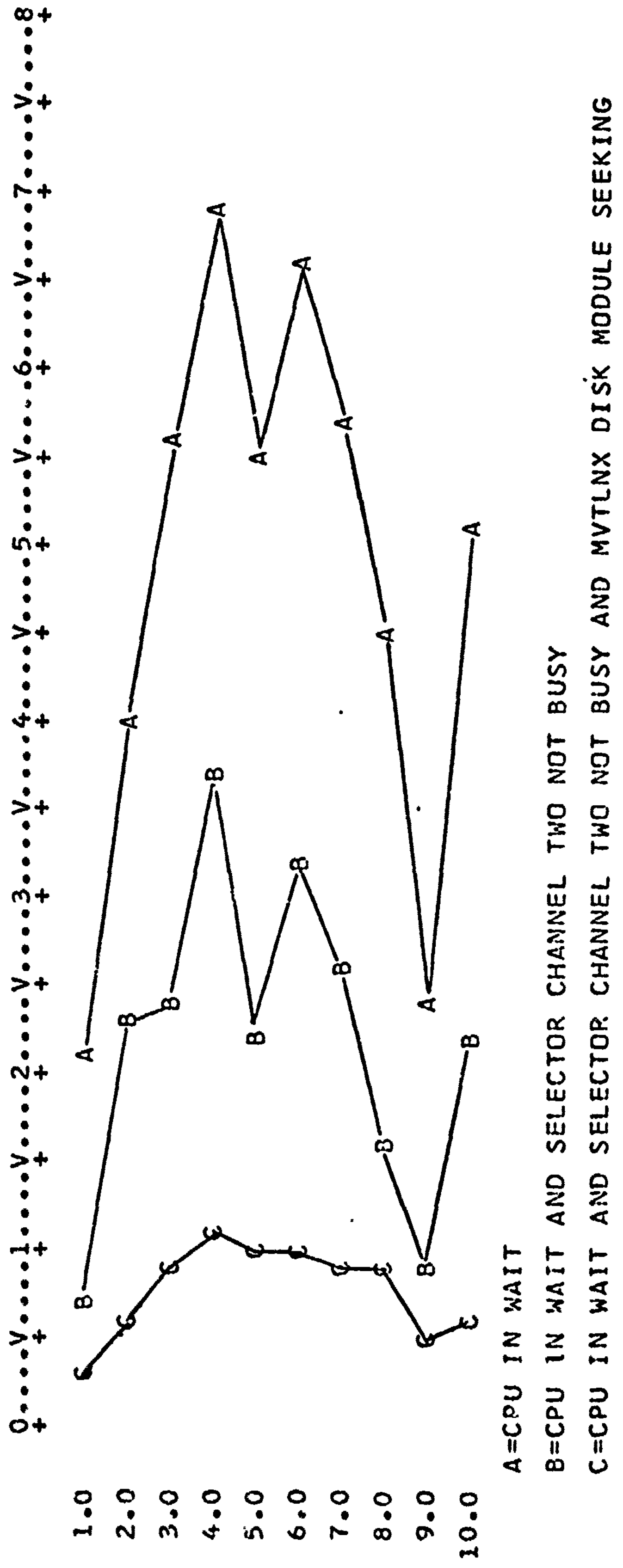
APPENDIX A
PLOTTING PROGRAM

```
//SPR10839 JOB (0839,0727FT,CS12),'SPRAGUE,J.W. SMC1286'
// EXEC FORTCLG
//FORT.SYSIN DD *
C
C     THE FIRST DIMENSION OF ARRAY P IS THE NUMBER OF DATA POINTS TO
C     BE PLOTTED FOR EACH CURVE; THE SECOND DIMENSION IS ONE GREATER
C     THAN THE NUMBER OF CURVES TO BE PLOTTED.
C
      DIMENSION P(26,4),SCALE(10)
      INTEGER*2 TITLE(60)
      DATA TITLE/60*0/,SCALE/10*0.0/
      DIMENSION STRING(20)
C
C     FIRST DATA CARD IS A TITLE CARD; THE 80 CHARACTERS ON THE CARD
C     WILL BE REPRODUCED ON THE FIRST PAGE OF OUTPUT, SEPERATE FROM
C     THE GRAPHS
C
      READ(5,444)(STRING(J),J=1,20)
  444 FORMAT(20A4)
      WRITE(6,333)(STRING(J),J=1,20)
  333 FORMAT(' ',20A4)
C
C     NNN MUST BE SET TO THE FIRST DIMENSION OF ARRAY P.
C
      NNN=26
      I=1
C
C     BY REWRITING THIS READ STATEMENT ADDITIONAL COUNTERS MAY BE PLOTTE
C    P(I,1) WILL ALWAYS BE THE LAST ELEMENT TO BE READ- IT IS THE
C     INTERVAL NUMBER.  THE INTERVAL NUMBER IS THE INDEPENDENT VARIABLE
C     AGAINST WHICH THE COUNTER VALUES ARE PLOTTED.
C
   10 READ(5,111,END=20)P(I,2),P(I,3),P(I,4),P(I,1)
C
C     THIS FORMAT STATEMENT MUST CONFORM TO THE PREVIOUS READ STATEMENT.
C     THE F2.0 CORRESPONDS TO THE INTERVAL NUMBER.
C
  111 FORMAT(7X,2F5.2,10X,F5.2,F2.0/)
      I=I+1
   20 CONTINUE
      LNCNT=I-1
C    NCURVE MUST BE THE NUMBER OF CURVES TO BE PLOTTED. IT USUALLY
C     WILL BE ONE LESS THAN THE SECOND DIMENSION OF ARRAY P.
      NCURVE=3
      WRITE(6,222)(I,P(I,2),I=1,LNCNT)
  222 FORMAT(' ',T25,'ECHO CHECK '/(' ',5(I4,5X,F5.2)))
      CALL STPLOT(NCURVE,LNCNT,TITLE,NNN,P,SCALE)
      STOP
      END                                                               MPLT0490

      SUBROUTINE STPLOT(NCUPVE,LNCNT,TITLE,NNN,P,SCALE)                 MPLT0500
      DIMENSION STRONG(20)                                              MPLT0510
      DIMENSION RANGE(10),SCALE(10),OFFSET(10),P(NNN,11),CONST(10)      MPLT0520
      REAL*4 MAX(10),MXTIME(10),MIN(10),MNTIME(10),LINE(103),ORIGIN (10)MPLT0530
      INTEGER*2 TITLE(60)                                               MPLT0540
      REAL PL(10)/4HA   ,4HB   ,4HC   ,4HD   ,4HE   ,4HF   ,            MPLT0550
     1           4HG   ,4HH   ,4HJ   ,4HI   /                           MPLT0560
      REAL DOT/4H.   /,BLANK/4H    /,PLUS/4H+   /                       MPLT0570
C                                                                       MPLT0580
C     INITIALIZING OF MAX,MXTIME,MIN,MNTIME                             MPLT0590
C                                                                       MPLT0600
      DO 1000 K=1,NCURVE                                                MPLT0610
      MAX(K)=-1.E75                                                     MPLT0620
      MXTIME(K)=0.                                                      MPLT0630
      MIN(K)=1.E75                                                      MPLT0640
 1000 MNTIME(K)=0.                                                      MPLT0650
C                                                                       MPLT0660
C     CALCULATE THE MAXIMUM AND MININUM VALUES FOR EACH INDEPENDENT     MPLT0670
C     VAPIABLE AS WELL AS THE FIRST TIME OF OCCURRENCE                  MPLT0680
C     NOTE: MAX(1) IS THE FIRST MAX FOR INDEPENDENT VARIABLE NUMBER 1   MPLT0690
C     STORED IN P(I,2).                                                 MPLT0700
C                                                                       MPLT0710
      DO 5 I=1,LNCNT                                                    MPLT0720
      DO 6 J=1,NCURVE                                                   MPLT0730
      IF(P(I,J+1)-MAX(J))7,7,8                                          MPLT0740
    8 MAX(J)=P(I,J+1)                                                   MPLT0750
      MXTIME(J)=P(I,1)                                                  MPLT0760
    7 CONTINUE                                                          MPLT0770
      IF(P(I,J+1)-MIN(J))9,6,6                                          MPLT0780
    9 MIN(J)=P(I,J+1)                                                   MPLT0790
      MNTIMF(J)=P(I,1)                                                  MPLT0800
    6 CONTINUE                                                          MPLT0810
    5 CONTINUE                                                          MPLT0820
      DO 27 I=1,NCURVE                                                  MPLT0830
      MAX(I)=99.99                                                      MPLT0840
      MIN(I)=0.0                                                        MPLT0850
   27 CONTINUE                                                          MPLT0860
C                                                                       MPLT0870
C                                                                       MPLT0880
C     DETERMINE IF EACH DEPENDENT VARIABLE IS A CONSTANT;(IF TRUE       MPLT0890
C     CONST(J)=0.)                                                      MPLT0900
C                                                                       MPLT0910
      DO 10 J=1,NCURVE                                                  MPLT0920
      CONST(J)=MAX(J)-MIN(J)                                            MPLT0930
C                                                                       MPLT0940
```

```
C     DETERMINE OFFSET IF THE DEPENDENT VARIABLE IS A CONSTANT.         MPLT0950
C                                                                       MPLT0960
      OFFSET(J)=0.                                                      MPLT0970
C                                                                       MPLT0980
C     DETERMINE OFFSET NECESSARY TO PLACE LOWEST VALUE OF DEPENDENT     MPLT0990
C     VARIABLE AT BOTTOM OF GRAPH IF THE VARIABLE IS NOT A CONSTANT.    MPLT1000
C                                                                       MPLT1010
      IF(CONST(J).NE.0.)OFFSET(J)=MIN(J)                                MPLT1020
   10 CONTINUE                                                          MPLT1030
C                                                                       MPLT1040
C                                                                       MPLT1050
C     DETERMINE THE RANGE OF EACH DEPENDENT VARIABLE.                   MPLT1060
C                                                                       MPLT1070
      DO 1 J=1,NCURVE                                                   MPLT1080
      RANGE(J)=MAX(J)-MIN(J)                                            MPLT1090
      IF(SCALE(J).NE.0.)RANGE(J)=10.*SCALE(J)                           MPLT1100
    1 CONTINUE                                                          MPLT1110
C                                                                       MPLT1120
C                                                                       MPLT1130
C     DETERMINE THE FINAL SCALE IN UNITS/INCH WHICH WILL BE PRINTED AS  MPLT1140
C     DATA BEFORE THE GRAPH.                                            MPLT1150
C                                                                       MPLT1160
      DO 104 J=1,NCURVE                                                 MPLT1170
      IF(RANGE(J).LT.1.E-68) GO TO 105                                  MPLT1180
      SCALE(J)=RANGE(J)/10.                                             MPLT1190
      IF(CONST(J).EQ.0.)GOTO 106                                        MPLT1200
C                                                                       MPLT1210
C     DETERMINE THE LOCATION OF THE ORIGIN (ZERO POINT) ON THE GRAPH    MPLT1220
C     FOR EACH DEPENDENT VARIABLE.                                      MPLT1230
C                                                                       MPLT1240
      ORIGIN(J)=(ABS(MIN(J))/SCALE(J))/10.                              MPLT1250
      GO TO 104                                                         MPLT1260
  105 SCALE(J)=0.                                                       MPLT1270
  106 ORIGIN(J)=0.                                                      MPLT1280
  104 CONTINUE                                                          MPLT1290
      WRITE(6,97)                                                       MPLT1300
   97 FORMAT('1')                                                       MPLT1310
C                                                                       MPLT1320
C    MAKE INITIAL LINE PRINTING .                                       MPLT1330
C                                                                       MPLT1340
      WRITE(6,94)                                                       MPLT1350
   94 FORMAT(19X,'0....V....1....V....2....V....3....V....4....V....5...MPLT1360
     1.V....6....V....7....V....8....V....9....V....0')                 MPLT1370
      WRITE(6,93)                                                       MPLT1380
   93 FORMAT(19X,'+   +   +   +   +   +   +   +   +   +   +             MPLT1390
     1 +   +   +   +   +   +   +   +   +   +')                          MPLT1400
      DO 200 LNC=1,LNCNT                                                MPLT1410
      DO 91 I=1,103                                                     MPLT1420
   91 LINE(I)=BLANK                                                      MPLT1430
C                                                                        MPLT1440
C     DETERMINE WHERE EACH DEPENDENT VARIABLE SYMBOL WILL BE PLOTTED.    MPLT1450
C                                                                        MPLT1460
      DO 89 J=1,NCURVE                                                   MPLT1470
      L=J+1                                                              MPLT1480
C                                                                        MPLT1490
C     MAKE RANGE(J)=1. IF THE VARIABLE IS A CONSTANT                     MPLT1500
C     AND AUTOSCALING IS NOT OVERRIDDEN.                                 MPLT1510
      IF(CONST(J).EQ.0..AND.RANGE(J).EQ.0.)RANGE(J)=1.                   MPLT1520
      SHIFT=P(LNC,L)-OFFSET(J)                                           MPLT1530
      IF(RANGE(J).EQ.0.) GOTO 86                                         MPLT1540
      K=100.*(SHIFT/RANGE(J))+2.5                                        MPLT1550
   86 CONTINUE                                                           MPLT1560
C                                                                        MPLT1570
C     THE NEXT 2 CARDS PREVENT OUTPUT FORMAT ERRORS FROM BEING GENERATEDMPLT1580
C     IF AUTOSCALING HAS BEEN OVERRIDDEN AND THE VALUE OF K COMPUTED     MPLT1590
C     FOR A VARIABLE IS GREATER THAN 102 OR LESS THAN 2. IF SUCH IS THE MPLT1600
C     CASE THE POINT WILL BE PLOTTED OUTSIDE THE GRAPH MARGINS OF +'S.   MPLT1610
C                                                                        MPLT1620
      IF(K.LT.2)K=1                                                      MPLT1630
      IF(K.GT.102)K=103                                                  MPLT1640
      IF(CONST(J).EQ.0..AND.SHIFT.LT.0.)K=1                              MPLT1650
      GOTO 85                                                            MPLT1660
   85 CONTINUE                                                           MPLT1670
      LINE(K)=RL(J)                                                      MPLT1680
   89 CONTINUE                                                           MPLT1690
      WRITE(6,747)                                                       MPLT1700
  747 FORMAT(' ')                                                        MPLT1710
      WRITE(6,88) P(LNC,1),LINE                                          MPLT1720
   88 FORMAT(12X,F4.1,2X,103A1)                                          MPLT1730
  200 CONTINUE                                                           MPLT1740
      WRITE(6,97)                                                        MPLT1750
      RETURN                                                             MPLT1760
      END                                                                MPLT1770
```

APPENDIX B
PLOTTING OUTPUT



APPENDIX C
CORRELATION MATRICES

```
BMD02D CORRELATION WITH TRANSGENERATION - REVISED JANUARY 29, 1970
HEALTH SCIENCES COMPUTING FACILITY,UCLA

PROBLEM CODE
NUMBER OF VARIABLES    9
NUMBER OF CASES      10

ARIABLE FORMAT CARD (S)
(7X,5F5.2/2X,4F5.2)

REMAINING SAMPLE SIZE=   10

SUMS

      484.4795    217.9199    265.9099    25.9200    79.2599    1.7700    55.7500    59.7100
       36.7600
MEANS

       48.4479     21.7920     26.5910     2.5920     7.9260    0.1770     5.5750     5.9710
        3.6760
STANDARD DEVIATIONS

       16.1555      9.1246      8.0146     1.0864     2.6097    0.4061    14.0239     4.8397
        9.1294
CORRELATION MATRIX

          COL.        COL.        COL.        COL.        COL.        COL.        COL.        COL.
           1           2           3           4           5           6           7           8
ROW

  1      1.0000      0.9468      0.9342      0.9520      0.9268      0.4558     -0.0330      0.4281
  2      0.9468      1.0000      0.7701      0.8662      0.8387      0.4642     -0.2024      0.2256
  3      0.9342      0.7701      1.0000      0.9293      0.9054      0.3927      0.1658      0.6020
  4      0.9520      0.8662      0.9293      1.0000      0.8779      0.6458      0.0234      0.4200
  5      0.9268      0.8387      0.9054      0.8779      1.0000      0.3556      0.1421      0.6151
  6      0.4558      0.4642      0.3927      0.6458      0.3556      1.0000     -0.0997      0.1206
  7     -0.0330     -0.2024      0.1658      0.0234      0.1421     -0.0997      1.0000      0.7411
  8      0.4281      0.2256      0.6020      0.4200      0.6151      0.1206      0.7411      1.0000
  9     -0.0356     -0.2046      0.1634      0.0104      0.1410     -0.1304      0.9984      0.7442
          COL.        COL.        COL.        COL.        COL.        COL.        COL.        COL.
           9
  1     -0.0356
  2     -0.2046
  3      0.1634
  4      0.0104
  5      0.1410
  6     -0.1304
  7      0.9984
  8      0.7442
  9      1.0000
```

APPENDIX D
FIGURE OF MERIT SURVEY

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Dear Sirs:

As part of a continuing study of computer performance measurement, a survey is being undertaken by the Naval Postgraduate School "Computer Science Group". This survey will seek to collect information about performance measurement at other computer installations. We are most interested in how performance is measured (hardware monitors, software monitors, accounting data), what parameters are measured (CPU utilization, I/O overlap, core utilization), what typical or realistic values for these parameters for particular job streams, and very importantly, what use is made of these results.

Your cooperation is requested in completing the enclosed form as completely and accurately as possible. The results of all returned surveys will be compiled and distributed to all contributors.

Sincerely,

G. H. SYMS
Assistant Professor

Installation Name Point of Contact

(1) Main Frame Designation/Model (2) Own/Lease/Rent

(3) Disk Units (model) (4)Number (5)Tape Drives (6)Number hours
of operation/day

(7) Core Storage (8) Amount (9) Bulk (slow) Core Stg. (10) Amount

(11) Drum (12) Capacity (13) Terminals (14)Number (15) Oper.
(time share) Systems

(16) Printers (17) Card Reader/Punch (18) Other Input/Ouput Devices

19. Size of user community? (Students, faculty, staff)

____

20. Job stream
    a. Jobs/month: ______________
    b. Job size distribution (core used): ____________
    c. Job time distribution (CPU time used): ________

21. Turnaround time
    a. Average per job: ______________
    b. Distribution: ______________

22. What type of performance measurements are implemented?

a. Hardware monitor?

1) Model:

2) Own/Lease:

3) Configuration:

No. probes:

No. accumulators:

(counters)

Recording media:

(mag tape, paper tape)

b. Software monitor?

1) Name:

2) Own/Lease:

3) Capabilities:

c. Accounting routines

1) Acquired from manufacturer/locally developed

23. Are the outputs of any measurement tools used as inputs to any type of configuration simulations? If so, which ones?

24. Just what parameters aremeasured in detail? Please give yes or no and recent mean values or ranges if possible.

a. CPU utilization

b. Channel utilization

c. Channel utilization while CPU wait

d. Device utilization

transfer
seek
queue length

e. Length of job queue

maximum
average

f. Core segment utilization

g. Overhead time (%)

h. System Data Sets

Transfer
Seek
Queue length

i. Supervisor Calls

Active
Loading
Inactive

Job Stream Data

j. Job arrival distribution

k. Distribution of jobs by language

l. Distribution of jobs by core size request

m. Distribution of jobs by time requested

n. Distribution of turnaround time by job time

o. Distribution of turnaround time by job size (core)

p. Amount of I/O per job

q. "Cost" per job (or charge schedule)

25. Is a full time time-sharing system supported?

What system?

26. If only a part time time-sharing is supported, during what hours of the day is it available?

27. Is a remote job entry capability supported?

28. Can the user monitor the queue status to determine where his job is located?

29. Are your"customers/users" satisfied with the performance of your computer system?

30. How do you know?

31. Are the staff/operators satisfied with the performance of computer system?

32. How do you know?

33. Is there any one overall figure of merit or performance index computered by combination of several performance parameters? (Please give formula.)

34. Which parameters in question 24 do you consider most significant as an indication of computer system performance?

LIST OF REFERENCES

1. Sewald, M. D., and others, "A Pragmatic Approach to Systems Measurement," Computer Decisions, p. 38-40 July 1971.

2. Bookman, P. G., Brotman, B. A. and Schmitt, K. L., "Use Measurement Engineering for Better System Performance," Computer Decisions, p. 28-32, April 1972.

3. Johnson, R. R., "Needed: A Measure for Measure," Datamation, v. 16, p. 22-30, 15 December 1970.

4. Hanke, R. R., Preliminary Steps in Optimizing University Computer Performance Using Hardware and Software Monitors, Masters Thesis, Naval Postgraduate School, Monterey, 1971.

5. UCLA Campus Computing Center Technical Report 4, The Design and Implementation of QUICKRUN, a Fast Job Management System Under OS/360 MVT, by R. H. Brubaker, Jr., and others, 26 February 1971.

6. Boole and Babbage, Incorporated, Measurement Engine Applications Handbook, 1971.

7. BIMED, Biomedical Computer Programs, Health Sciences Computing Facility, University of California, Los Angeles, 1964.

8. Estrin, G., Muntz, R. R., and Uzgalis, R. C., Modeling, Measurement and Computer Power, a paper presented at the Spring Joint Computer Conference, Atlantic City, New Jersey, May 1972.